Database Table Editor

Test plan

Version 1

April 8, 2020

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

|  |  |
| --- | --- |
| Initial Release: | April 8, 2020 |
| Current Release: | April 10, 2020 |
| Indicator of Last Page in Document: | \*\* |
| Date of Last Review: |  |
| Date of Next Review: |  |
| Target Date for Next Update: |  |

Distribution List

This following list of people shall receive a copy of this document every time a new version of this document becomes available:

Guidance Team Members: Dr. Roach

Team Members: Andrea Torres, Julio De La Cruz, Kevin Honsaker

Change Summary

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1 | April 8, 2020 | Andrea Torres | Initial Release of Document |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Document Control ii](#_Toc37454709)

[Approval ii](#_Toc37454710)

[Document Change Control ii](#_Toc37454711)

[Distribution List ii](#_Toc37454712)

[Change Summary ii](#_Toc37454713)

[1. Introduction 1](#_Toc37454714)

[1.1. Purpose 1](#_Toc37454715)

[1.2. Scope 1](#_Toc37454716)

[1.3. System Overview 1](#_Toc37454717)

[1.4. Suspension and Exit Criteria 1](#_Toc37454718)

[1.5. Document Overview 1](#_Toc37454719)

[1.6. References 1](#_Toc37454720)

[2. Test Items and Features 2](#_Toc37454721)

[3. Testing Approach 3](#_Toc37454722)

[4. Test 01 4](#_Toc37454723)

[4.1. Test <<test id>> 4](#_Toc37454724)

[5. Test 02 2](#_Toc37454725)

[5.1. Test <<test id>> 2](#_Toc37454726)

[6. Test 03 2](#_Toc37454727)

[6.1. Test <<test id>> 2](#_Toc37454728)

[7. Test 04 2](#_Toc37454729)

[7.1. Test <<test id>> 2](#_Toc37454730)

[8. User Interface Testing 3](#_Toc37454731)

[9. Test Schedule 4](#_Toc37454732)

[10. Other Sections 5](#_Toc37454733)

[11. Appendix 6](#_Toc37454734)

# Introduction

<< This section gives introductory information regarding the project, the system to be tested, and the testing approach.>>

## Purpose

<< Identify the project and stipulate the test plan purpose by indicating what the document contains (e.g., organizational responsibilities, test approach, test schedule. There are generally four different types of test plans: project test plan that describes the overall strategy for testing; the system test plan that describes the system from the customer’s point of view; integration test plan that describes integration of units and subsystems; unit test plan that describes modules or classes. This section needs to identify which of these this document is.>>

## Scope

<<Specify the project software releases/versions encompassed by the plan. >>

## System Overview

<<Describe the system to be exercised by the testing approach specified in the plan. This overview serves to identify aspects of the system operation that will be the focus of the plan’s testing approach. This should align with the systems overview of other documents in the project.>>

## Suspension and Exit Criteria

<< “suspension criteria” describes when we suspend testing, to be resumed at a later time. For example, if 40% of the test cases fail, or if any of the critical test cases fail. If there are no suspension criteria, indicate that all tests cases will be executed. “Exit criteria” indicates when testing stops. This could be based on run rate (number of test cases run divided by number of test cases specified) or pass rate (number of test cases passed divided by number of test cases run, or test cases passed divided by number of test cases specified). Nominally, we expect to run all of the specified tests. We want the pass rate to be high. We might specify that all critical tests must pass, and 90% of the non-critical must pass. In general, we want this to be high. >>

## Document Overview

<<Describe the remainder of the document.>>

## References

<<List all the references applicable to the test plan. Generally, this includes project standards, SRS, SDD, and a product assurance plan.>>

# Test Items and Features

<< This section describes the test items (e.g., components, classes, functions or methods) and the features to be tested. It may also list features not to be tested. A class diagram is useful. A table of features is useful. >>

# Testing Approach

<<Describe the approach to be used to the test the system. This description includes specifying the types of tests to be performed, e.g., tests designed to exercise system functions one by one; tests designed to exercise sequences of functions that approximate operational use of the system; tests designed to stress the system to its design and requirements limits. The description lists the specific tests to be performed but does not give the test steps. For each of these tests, give it a name and specify its objective. Label the criticality of the test cases. >>

Table 1: Test Plan

|  |  |  |
| --- | --- | --- |
| **TEST SUITE DBCompDup** | | |
| **Description of Test Suite** | This test suite will befocused on the ability to compare files to other versions and check for duplicate rows | |
| **Test Case Identifier** | **Objective** | **Criticality** |
| DBComp1 |  |  |
| DBComp2 |  |  |
| DBDup1 |  |  |
| DBDup2 |  |  |
|  |  |  |

# Test 01

<<The purpose of this section is to:

* document test input, specific test procedures, and outcomes.
* establish test methods,
* explain the nature and extent of each test >>

<< for each test case, complete the following: >>

## Test DBComp1

**Objective:**

**Notes:** <<This area provides general notes concerning the test procedure. Such notes might include comments on how to execute the test procedure, an estimate of the test duration, the requirements of the procedure tests, or a statement of resources needed for this test.>>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: DBComp1 | | | | Current Status: Pending | | |
| Test title: Data Base Comparison one. | | | | | | |
| Testing approach: <<Included in this section is a description of test harnesses, testing frameworks, environmental requirements, test tools and test automation that will be employed to achieve testing. Include naming conventions for tests and test scripts if appropriate. Provide requirements traceability and test priority.  >> | | | | | | |
| STEP  <<N>> | OPERATOR ACTION  Describe the actions taken by the person executing the test procedure. Include the test suite, or the name of the test file (in this case, the contents of the file should be given in the appendix). | PURPOSE  Describe the reason for the step. | | | EXEPCTED RESULTS  Describe the expected response of the system being tested to the action specified under OPERATOR ACTION. This should be derived from the SRS and SDD. Clearly indicate how we determine whether the step passes. | COMMENTS |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

# Test 02

<<The purpose of this section is to:

* document test input, specific test procedures, and outcomes.
* establish test methods,
* explain the nature and extent of each test >>

<< for each test case, complete the following: >>

## Test DBComp2

**Objective: <**< Define the objective of Test XX.Y. >>

**Notes:** <<This area provides general notes concerning the test procedure. Such notes might include comments on how to execute the test procedure, an estimate of the test duration, the requirements of the procedure tests, or a statement of resources needed for this test.>>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: DBComp2 | | | | Current Status: << Passed / Failed / Pending >> | | |
| Test title: <<This line contains the long title of the test procedure.>> | | | | | | |
| Testing approach: <<Included in this section is a description of test harnesses, testing frameworks, environmental requirements, test tools and test automation that will be employed to achieve testing. Include naming conventions for tests and test scripts if appropriate. Provide requirements traceability and test priority.  >> | | | | | | |
| STEP  <<N>> | OPERATOR ACTION  Describe the actions taken by the person executing the test procedure. Include the test suite, or the name of the test file (in this case, the contents of the file should be given in the appendix). | PURPOSE  Describe the reason for the step. | | | EXEPCTED RESULTS  Describe the expected response of the system being tested to the action specified under OPERATOR ACTION. This should be derived from the SRS and SDD. Clearly indicate how we determine whether the step passes. | COMMENTS |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

# Test 03

<<The purpose of this section is to:

* document test input, specific test procedures, and outcomes.
* establish test methods,
* explain the nature and extent of each test >>

<< for each test case, complete the following: >>

## Test DBDup1.1

**Objective:** The goal of Test 03.1 is to make sure that when there are no duplicates then none of the rows are highlighted yellow.

**Notes:** <<This area provides general notes concerning the test procedure. Such notes might include comments on how to execute the test procedure, an estimate of the test duration, the requirements of the procedure tests, or a statement of resources needed for this test.>>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: DBDup1.1 | | | | Current Status: Pending | | |
| Test title: Database Duplicate No Duplicates | | | | | | |
| Testing approach: For this test we will be using the Dup\_DB\_No\_Dups\_Table, this will be done manually. To conduct this test first open the dbEdit.jar and then run Dup\_DB file. No\_Dups\_Table. This will open the No\_Dups\_Table, it is observeable that there are no duplicates between any row. | | | | | | |
| STEP  1  2 | OPERATOR ACTION  Run dbEdit.jar open Dup\_DB, select No\_Dups\_Table  View the table makesure that no row is colored yellow. | PURPOSE  This step is to start the test properly.  To visually verify that the software did not mistakenly highlight a row. | | | EXEPCTED RESULTS  No\_Dups\_Table is opened. The table does not have any duplicate rows.  No rows are highlighted yellow because the table does not have any duplicates. | COMMENTS |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

## Test DBDup1.2

**Objective:** The goal of Test 03.2 is to make sure that when there are no duplicates then none of the rows are highlighted yellow when a nonduplicate row is added.

**Notes:** <<This area provides general notes concerning the test procedure. Such notes might include comments on how to execute the test procedure, an estimate of the test duration, the requirements of the procedure tests, or a statement of resources needed for this test.>>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: DBDup1 | | | | Current Status: Pending | | |
| Test title: Database Duplicate No Duplicates add a row that is not a duplicate. | | | | | | |
| Testing approach: For this test we will be using the Dup\_DB\_No\_Dups\_Table, this will be done manually. To conduct this test first open the dbEdit.jar and then run Dup\_DB file. No\_Dups\_Table. This will open the No\_Dups\_Table, it is observable that there are no duplicates between any row. The tester will manually add a row that does not have a duplicate. The table has 3 columns if something was to go wrong when inputting new information it should not fail in the first middle or last columns. After each entry the row should still be unique and there should be no yellow highlighting because all the rows are unique and not duplicates of each other. | | | | | | |
| STEP  1  2  3  4  5  6 | OPERATOR ACTION  Run dbEdit.jar open Dup\_DB, select No\_Dups\_Table  View the table make sure that no row is colored yellow.  Select “Edit” from the top tool bar. Select “Insert Row” from the dropdown menu.  Populate the first column “First Name” with “Jane”.  Populate the second column “Last Name” with “Doe”.  Populate the third column “Age” with 37 | PURPOSE  This step is to start the test properly.  To visually verify that the software did not mistakenly highlight a row.  To create a new empty row to later in the test populate.  To populate the first cell in the new row.  To populate the middle cell in the new row.  To populate the last cell in the new row. | | | EXEPCTED RESULTS  No\_Dups\_Table is opened. The table does not have any duplicate rows.  No rows are highlighted yellow because the table does not have any duplicates.  A new row is added, the row is not highlighted yellow because the empty row is unique.  The row is still unique no row is highlighted yellow.  The row is still unique no row is highlighted yellow.  The row is still unique no row is highlighted yellow. | COMMENTS |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

## Test DBDup1.3

**Objective:** The goal of Test 03.3 is to make sure that when there are no duplicates then none of the rows are highlighted yellow. In This test we will add a row that will become a duplicate.

**Notes:** <<This area provides general notes concerning the test procedure. Such notes might include comments on how to execute the test procedure, an estimate of the test duration, the requirements of the procedure tests, or a statement of resources needed for this test.>>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: DBDup1 | | | | Current Status: Pending | | |
| Test title: Database Duplicate No Duplicates | | | | | | |
| Testing approach: For this test we will be using the Dup\_DB\_No\_Dups\_Table, this will be done manually. To conduct this test first open the dbEdit.jar and then run Dup\_DB file. No\_Dups\_Table. This will open the No\_Dups\_Table, it is observeable that there are no duplicates between any row. | | | | | | |
| STEP  1  2 | OPERATOR ACTION  Run dbEdit.jar open Dup\_DB, select No\_Dups\_Table  View the table makesure that no row is colored yellow. | PURPOSE  This step is to start the test properly.  To visually verify that the software did not mistakenly highlight a row. | | | EXEPCTED RESULTS  No\_Dups\_Table is opened. The table does not have any duplicate rows.  No rows are highlighted yellow because the table does not have any duplicates. | COMMENTS |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

# Test 04

<<The purpose of this section is to:

* document test input, specific test procedures, and outcomes.
* establish test methods,
* explain the nature and extent of each test >>

<< for each test case, complete the following: >>

## Test DBDup2

**Objective: <**< Define the objective of Test XX.Y. >>

**Notes:** <<This area provides general notes concerning the test procedure. Such notes might include comments on how to execute the test procedure, an estimate of the test duration, the requirements of the procedure tests, or a statement of resources needed for this test.>>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test No.: DBDup2 | | | | Current Status: << Passed / Failed / Pending >> | | |
| Test title: <<This line contains the long title of the test procedure.>> | | | | | | |
| Testing approach: <<Included in this section is a description of test harnesses, testing frameworks, environmental requirements, test tools and test automation that will be employed to achieve testing. Include naming conventions for tests and test scripts if appropriate. Provide requirements traceability and test priority.  >> | | | | | | |
| STEP  <<N>> | OPERATOR ACTION  Describe the actions taken by the person executing the test procedure. Include the test suite, or the name of the test file (in this case, the contents of the file should be given in the appendix). | PURPOSE  Describe the reason for the step. | | | EXEPCTED RESULTS  Describe the expected response of the system being tested to the action specified under OPERATOR ACTION. This should be derived from the SRS and SDD. Clearly indicate how we determine whether the step passes. | COMMENTS |
| Concluding Remarks: | | | | | | |
| Testing Team:  << List members of testing team and lead >> | | | Date Completed: | | | |

# User Interface Testing

<<This section focuses on the interaction between the user and the system. For testing the user interface, consider the following traits:

* Consistent terminology, shortcut keys, menu selections, and presentation
* Correct language, spelling, and grammar.
* Flexibility in navigation between windows and interface elements.
* Error handling that will inform user of critical operations.
* Follows standards and guidelines such as placement of scroll bars, windows, and menu items.

This section could be integrated into Section 4.

>>

# Test Schedule

<< Specify the schedule for testing activities. A table with the order and completion dates of the tests is useful. The table below might be useful.>>

|  |  |  |
| --- | --- | --- |
| **Task and date** | **People** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Other Sections

<< Other sections that may appear in a test plan (but not required for this course) are:

* Test Management Requirements: how testing is to be managed; a delineation of responsibilities of each project organization involved with testing
* Staffing and training needs: delineate the responsibilities of those individuals who are to perform the testing, level of skill required, and training to be provided
* Environmental Requirements: describe the hardware (including communication and network equipment) needed to support testing; describe configuration of hardware components on which software and database to be tested are to operate.
* Software Requirements: describe the software needed to support testing; include the software code and databases that are object of the testing. Also include software tools such as compilers, CASE instruments and simulators that are needed to model the user’s operational environment.
* Risk and contingencies
* Cost: include an estimate of costs.
* Approvals
* Test Deliverables

>>

# Appendix

<< possibly more readable to put the expected output here and refer to it in the previous sections. Might also provide explicit directions for analysis of output, if it’s easier to read as an appendix or if analysis is post execution. >>